

REMARKS

Claims 1-27 were pending as of the date of the Office Action. Claims 1, 3, 4, 7-9, 13, 15, 18, 20 and 25 have been amended. Claims 2, 10-12, 14, 17, 19, 21-24, 26 and 27 have been canceled. Support for the amendments may be found through the specification, and no new matter has been added.

Prior to the amendments made above, claims 13-14, 17-22 and 24-27 stood rejected under 35 U.S.C. 102(b) as being anticipated by O'Leary (US 5,950,000). Claims 1-2 and 7-12 stood rejected under 35 U.S.C. 103(a) as being unpatentable over O'Leary in view of Mason (US Patent 6,817,005), and claims 3-6, 15-16 and 23 stood rejected under Section 103 as being unpatentable over O'Leary and Mason and further in view of Odaka (US 2003/0140333). Reconsideration is respectfully requested.

Independent claim 1 has been amended to recite:

“a tool file ... comprising a custom build rule for the tool that includes *a generalized description of the syntax of a command line that is run when the tool is executed*, the generalized description of the syntax of the command line including *at least one tag identifying a dynamic property of a command line switch* of the command line,”

a generic property store ... storing values for the dynamic property of the command line switch; and

a content handler executing on the processor that receives the tool file and generates from the custom build rule a custom build rule object, the *custom build rule object evaluating the dynamic property of the command line switch and automatically replacing the tag in the generalized description of the command line with one or more values from the generic property store* to transform the generalized description of the command line into an executable command line—comprising the name of the tool to be executed and one or more command line switch properties associated with the tool.”

(emphasis added). Independent claims 13 and 25 have been amended to recite similar features. None of the cited references teaches or suggests at least the highlighted features.

The Office Action seems to suggest that O'Leary teaches the highlighted features in column 6, lines 3-25, 38-45 and 46-65, however, it does not. The cited portions of the O'Leary reference describe how a user *manually* selects items from a menu of an “IPE Manager 200” in order to create a “Makefile” for an application (“app”) created by a “Visual” tool in an integrated design environment. The “Makefile” is simply a file that is used with a “make tool 206” that is part of the design environment. Importantly, however,

the “make tool 206” is one of several “selected integrated tools 201 of [the] integrated development environment” that is “‘bundled’ with IPE Manager 200 which has been informed *a priori* of the capabilities of the [tool]” (col. 4, ll. 48-53) (emphasis added). Thus, the “make tool 206” on which the Office Action tries to draw analogies is not a build tool that needs to be integrated into the design environment of O’Leary. Rather, as O’Leary describes, the “make tool 206” is already an integrated tool of the IPE Manager 200 and one whose capabilities the IPE Manager 200 is already aware of.

On the contrary, the claimed system and method enables a tool that is not part of an integrated design environment to be able to be integrated into that environment in a manner that is transparent to the user. In order to do so, a tool file containing a custom build rule for the tool is created. That custom build rule includes a generalized description of the syntax of a command line that is run when the tool is executed. The generalized description of the command line includes at least one tag identifying a dynamic property of a command line switch of the command line. A content handler then receives the tool file and generates a custom build rule object from the custom build rule in the file. The custom build rule object *evaluates* the dynamic property of the command line switch and *automatically replaces* the tag in the generalized description of the command line with one or more values from a generic property store to transform the generalized description of the command line into an executable command line. O’Leary does not describe such a system.

In particular, there is no description in O’Leary of generating a custom build rule object from a custom build rule in a tool file. Nor is there any description of that custom build rule object “*evaluating the dynamic property* of [a] command line switch and *automatically replacing* [a] tag in [a] generalized description of [a] command line [for the tool] with one or more values from [a] generic property store.” The Office Action seems to suggest that parts of the pathname of a Makefile are analogous to dynamic properties of a command line switch, as claimed, but that is not the case. Indeed, there is no description in O’Leary of “evaluating the dynamic property [a] command line switch,” nor any description of a custom build rule object “automatically replacing the tag in the generalized description of the command line with one or more values from the generic property store.” On the contrary, the pathname of the Makefile in O’Leary appears to be selected by a user from a

menu. Thus, at least the foregoing features are missing from the O'Leary reference. Nor do the other cited references cure the deficiencies of O'Leary.

Because the foregoing features of independent claims 1, 13 and 25 are neither taught nor suggested by O'Leary, Mason or Odaka, whether alone or in combination, the applicants respectfully submit that those claims are patentable over the cited references. Inasmuch as the remaining claims all depend from one of those independent claims, they too are patentable for the same reasons. Reconsideration of the Section 102 and 103 rejections is respectfully requested.

CONCLUSION

For all the foregoing reasons, the applicants respectfully submit that the instant application is in condition for allowance.

Date: May 19, 2009

/Steven B. Samuels/

Steven B. Samuels

Registration No. 37711

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439